Homework 1: Searching and Sorting

Due: September 18, 2025 at 2:30p.m.

This homework must be typed in LaTeX and submitted via Gradescope.

Please ensure that your solutions are complete, concise, and communicated clearly. Use full sentences and plan your presentation before your write. Except where indicated, consider every problem as asking for a proof.

Problem 1. The *Fibonacci numbers* is a sequence of numbers starting with $f_1 = f_2 = 1$ defined by the recurrence:

$$f_{n+2} = f_n + f_{n+1}$$

for $n \geq 1$. Use induction to prove the following formula for $n \geq 1$:

$$P(n) := \sum_{i=1}^{n} f_i^2 = f_n f_{n+1}$$

Solution. □

Problem 2. A *geometric sequence* with common ratio r is a sequence of numbers given by:

$$a_1,\ a_1 r,\ a_1 r^2,\ a_2 r^3,\ \cdots$$

For example, the following is a geometric sequence with a common ratio 2.

$$1,\ 2,\ 4,\ 8,\ 16,\ \cdots$$

(a) Describe an algorithm to find the value of a deleted term of a geometric sequence of length n with common ratio r in $O(\log n)$ time. For example, the sequence

$$1,\ \frac{1}{3},\ \frac{1}{9},\ \frac{1}{81},\ \frac{1}{243}$$

is missing the term $\frac{1}{27}$.

(b) Provide a succinct proof of the correctness of the algorithm.

(c) Provide an analysis of the running time (asymptotic analysis is sufficient).

Solution. □

Problem 3. Let $X = [a_1, \cdots, a_n]$ and $Y = [y_1, \cdots, y_n]$ be two sorted arrays (in non-decreasing order). For simplicity, assume n is a power of 2.

(a) Describe an algorithm to find the median of all $2n$ elements in the arrays X and Y in $O(\log n)$ time.

(b) Provide a succinct proof of the correctness of the algorithm.

(c) Provide an analysis of the running time (asymptotic analysis is correct) and memory utilization of the algorithm.

Hint: Note that the given arrays are already sorted and of the **same size**! You may want to use binary search to exploit this fact. Remove the $n/2$ elements that are smaller than the medians of both arrays and the $n/2$ elements that are greater than both medians and iterate on the remainig set.

Solution. □

Problem 4. Let S be an array of n *distinct* integers. An inversion in S is a pair of indices i and j such that $i < j$, but $S_i > S_j$. For example, the following sequence has six inversions:

$$\{8,\ 6,\ 4,\ 1\}$$
$$(8,6),\ (8,4),\ (8,1),\ (6,4),\ (6,1),\ (4,1)$$

(a) Provide a succinct (but clear) description of an algorithm running in $O(n \log n)$ time to determine the number of inversions in S. You may provide a pseudocode.

(b) Provide a succinct proof of the correctness of the algorithm.

(c) Provide an analysis of the running time (asymptotic analysis is correct) and memory utilization of the algorithm.

Solution. □